APPROVED O.G. FIG.
5Y
DRAFTSMAN

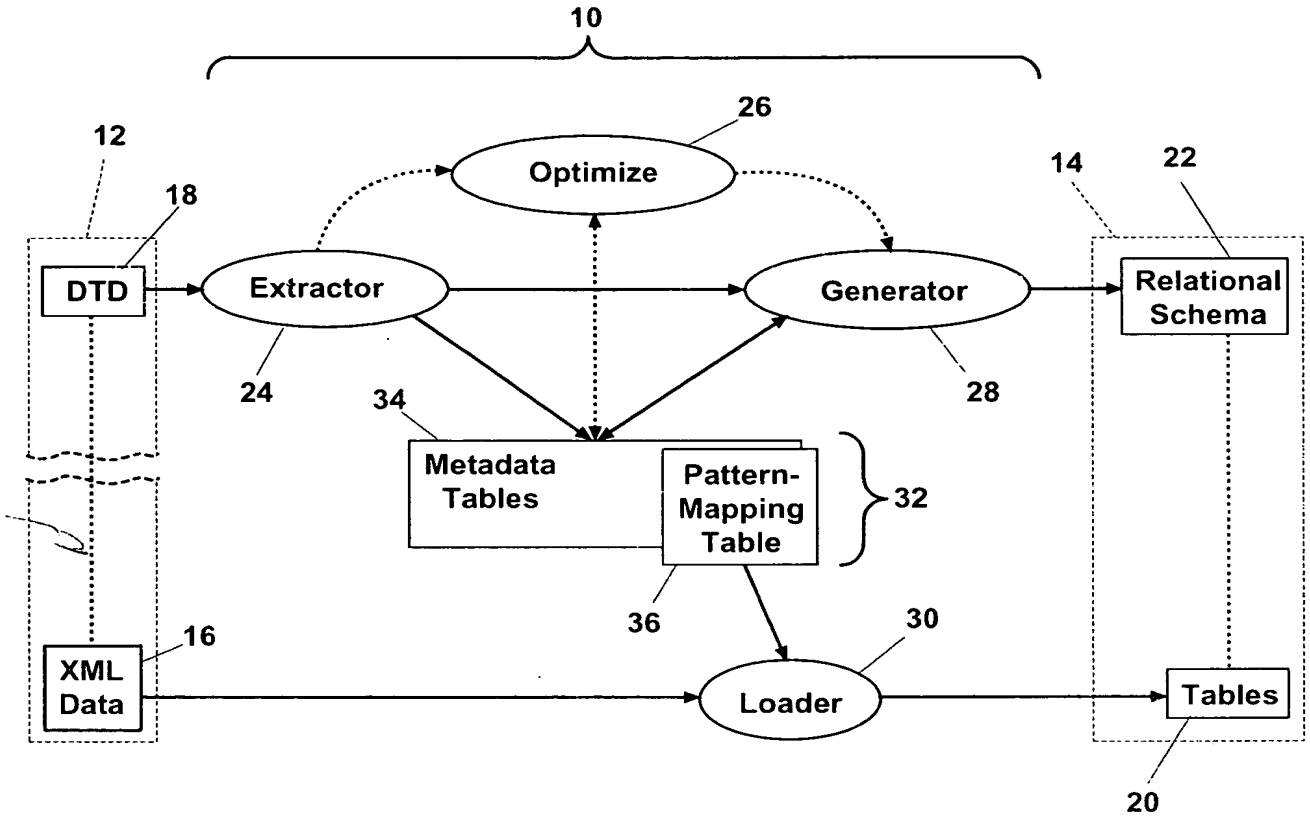

Fig. 1

FOHF20" 2598760

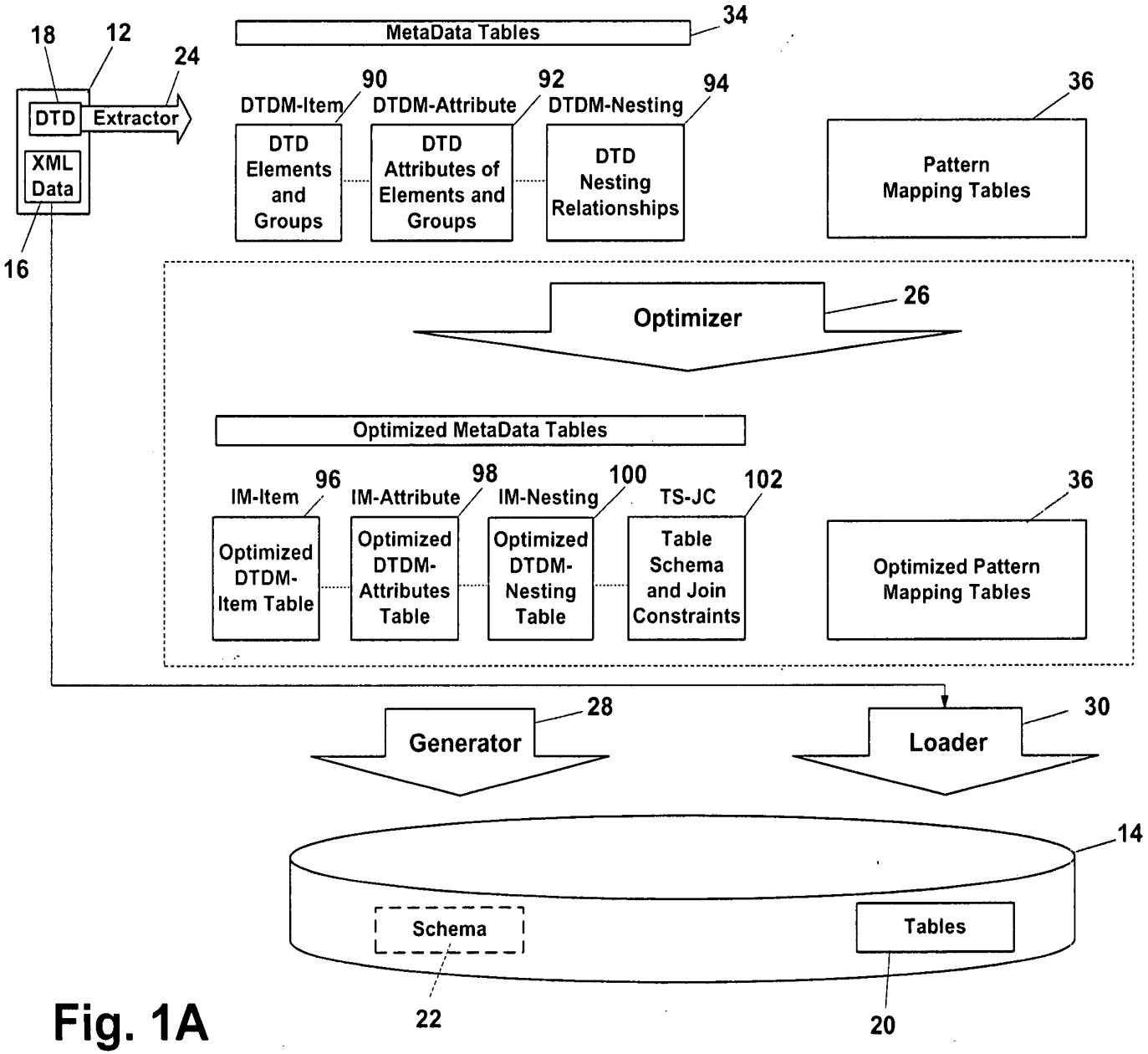

Fig. 1A

APPROVED O.G. FIG.

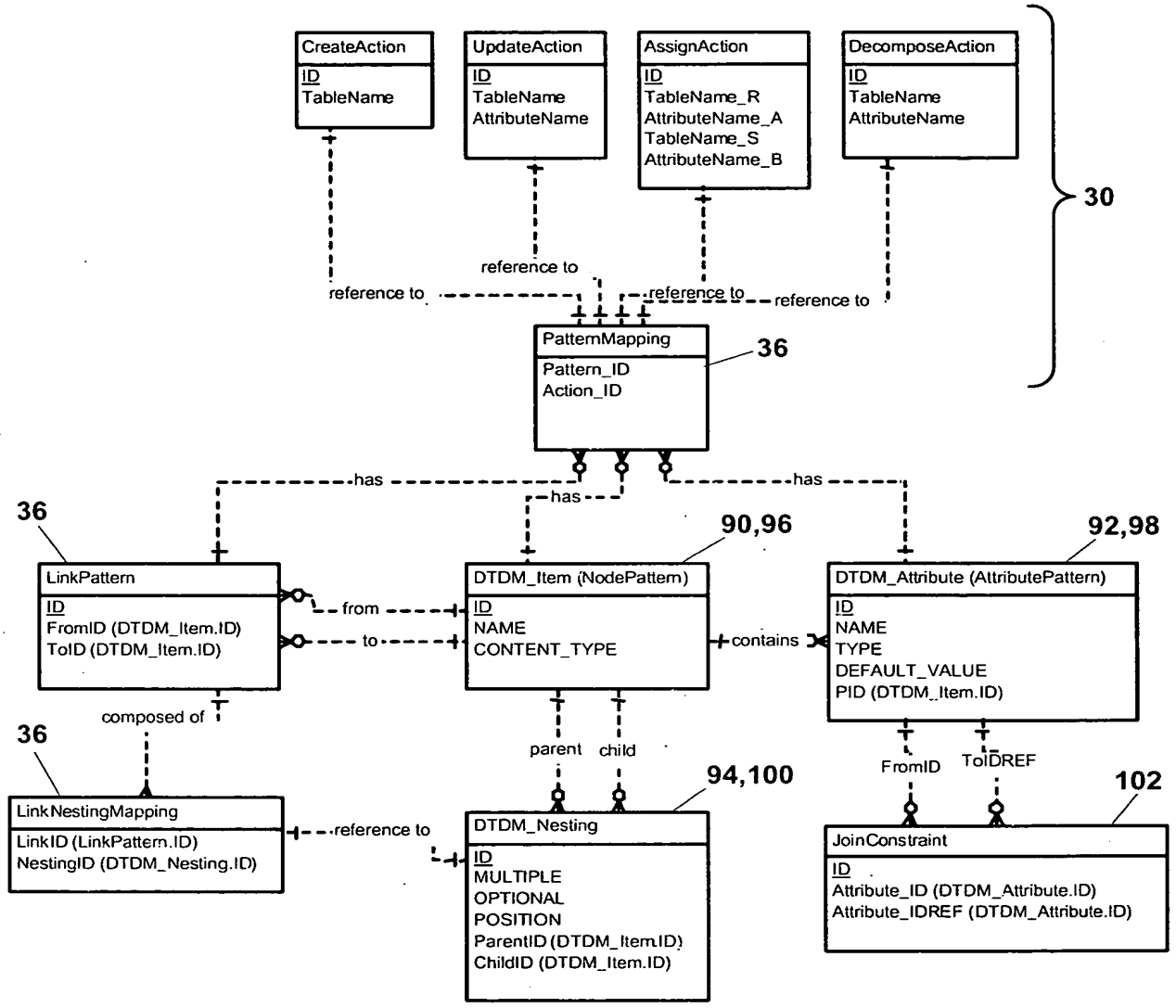

Fig. 1B

APPROVED

Patent: 7,685,519

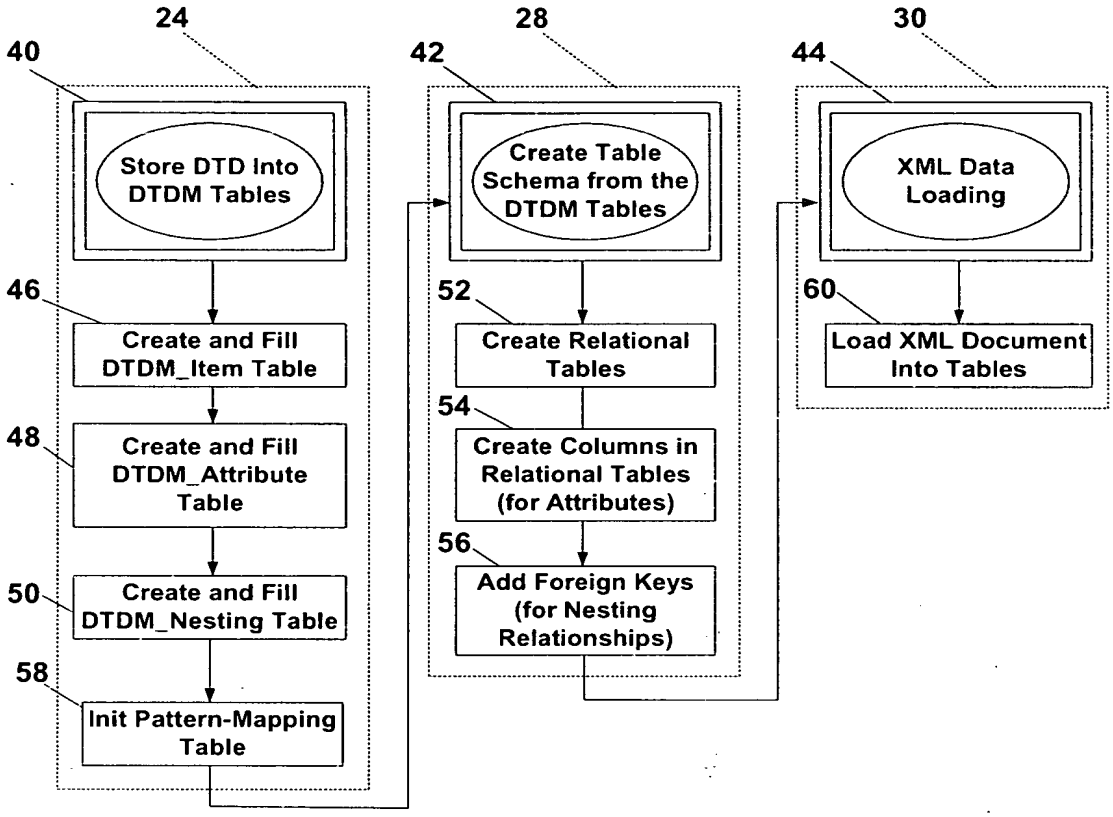

Fig. 2

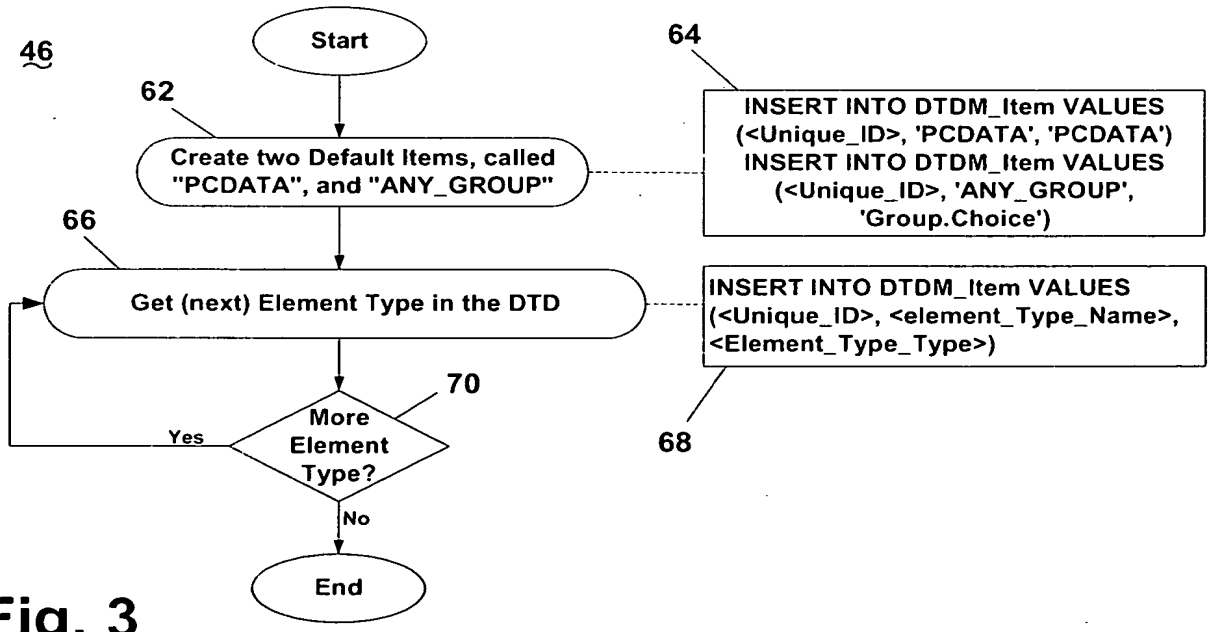

Fig. 3

APPROVED O.G. FIG.

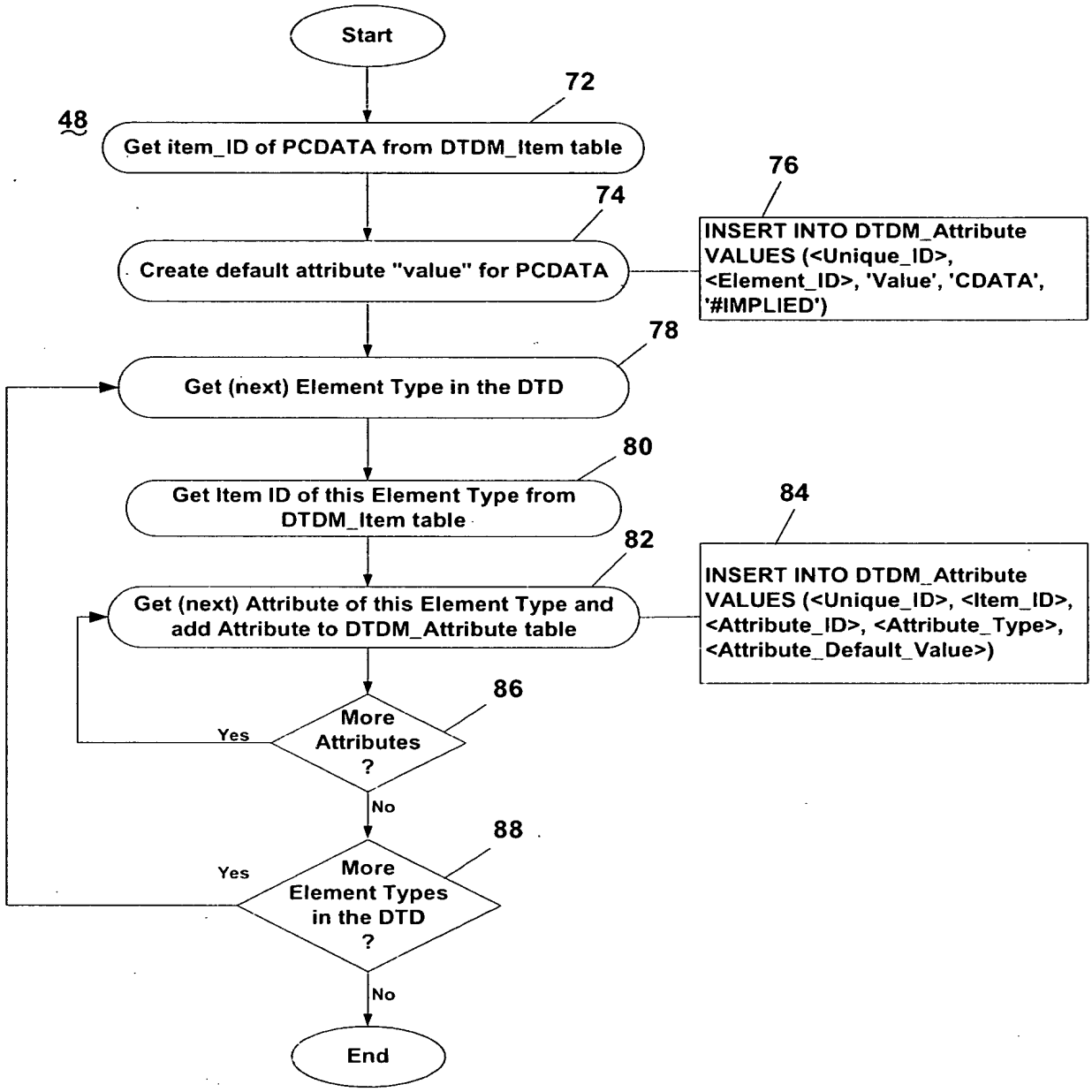

Fig. 4

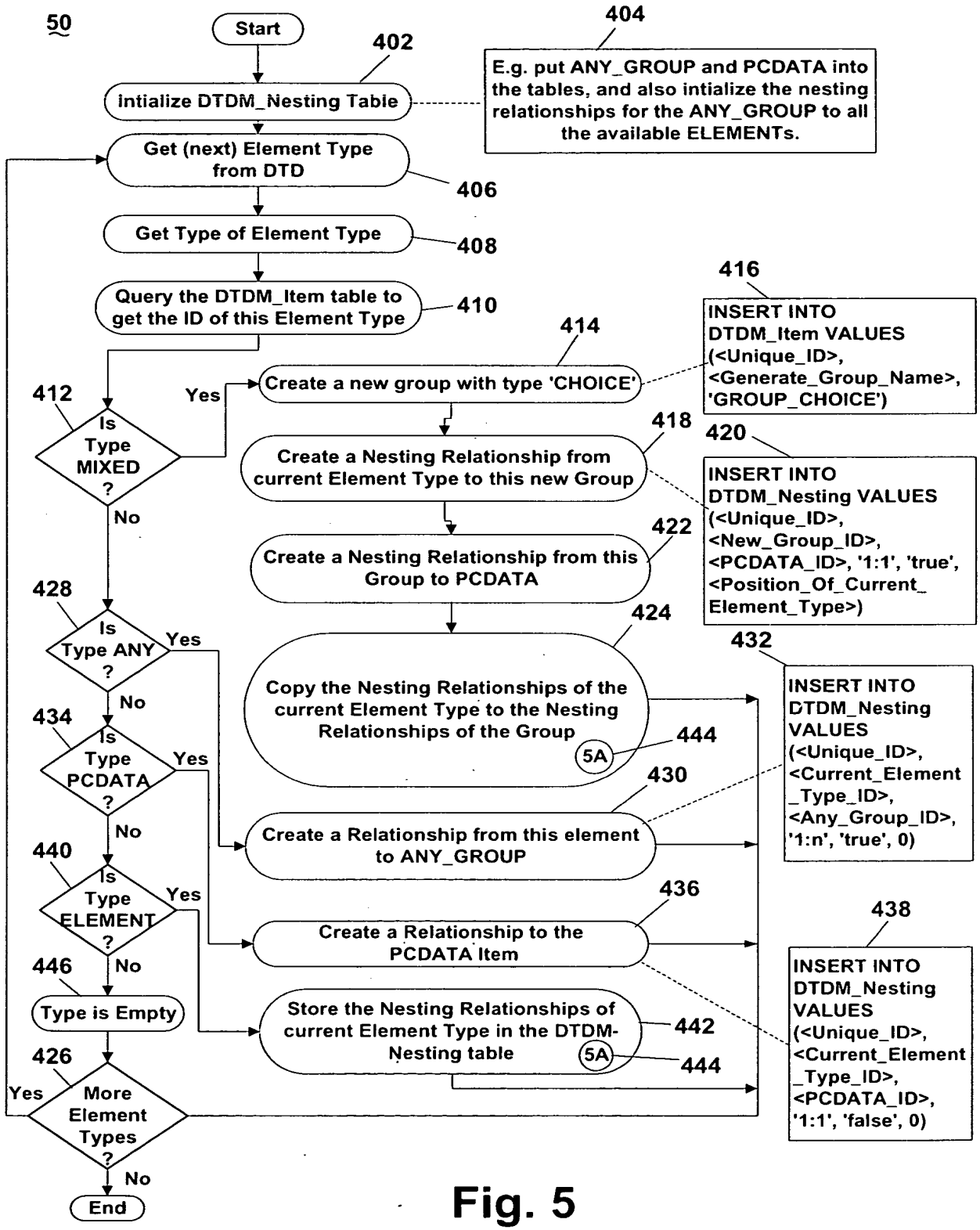
50
Start
402
Intialize DTDM_Nesting Table
404
E.g. put ANY_GROUP and PCDATA into the tables, and also intialize the nesting relationships for the ANY_GROUP to all the available ELEMENTs.
Get (next) Element Type from DTD
406
Get Type of Element Type
408
Query the DTDM_Item table to get the ID of this Element Type
410
412
Is Type MIXED ?
Yes
414
Create a new group with type 'CHOICE'
416
INSERT INTO DTDM_Item VALUES (<Unique_ID>, <Generate_Group_Name>, 'GROUP_CHOICE')
418
Create a Nesting Relationship from current Element Type to this new Group
420
INSERT INTO DTDM_Nesting VALUES (<Unique_ID>, <New_Group_ID>, <PCDATA_ID>, '1:1', 'true', <Position_Of_Current_Element_Type>)
No
422
Create a Nesting Relationship from this Group to PCDATA
428
Is Type ANY ?
Yes
424
Copy the Nesting Relationships of the current Element Type to the Nesting Relationships of the Group
5A
444
432
INSERT INTO DTDM_Nesting VALUES (<Unique_ID>, <Current_Element_Type_ID>, <Any_Group_ID>, '1:n', 'true', 0)
No
434
Is Type PCDATA ?
Yes
430
Create a Relationship from this element to ANY_GROUP
No
440
Is Type ELEMENT ?
Yes
436
Create a Relationship to the PCDATA Item
438
INSERT INTO DTDM_Nesting VALUES (<Unique_ID>, <Current_Element_Type_ID>, <PCDATA_ID>, '1:1', 'false', 0)
No
446
Type is Empty
Store the Nesting Relationships of current Element Type in the DTDM-Nesting table
442
5A
444
426
Yes
More Element Types ?
No
End

Fig. 5

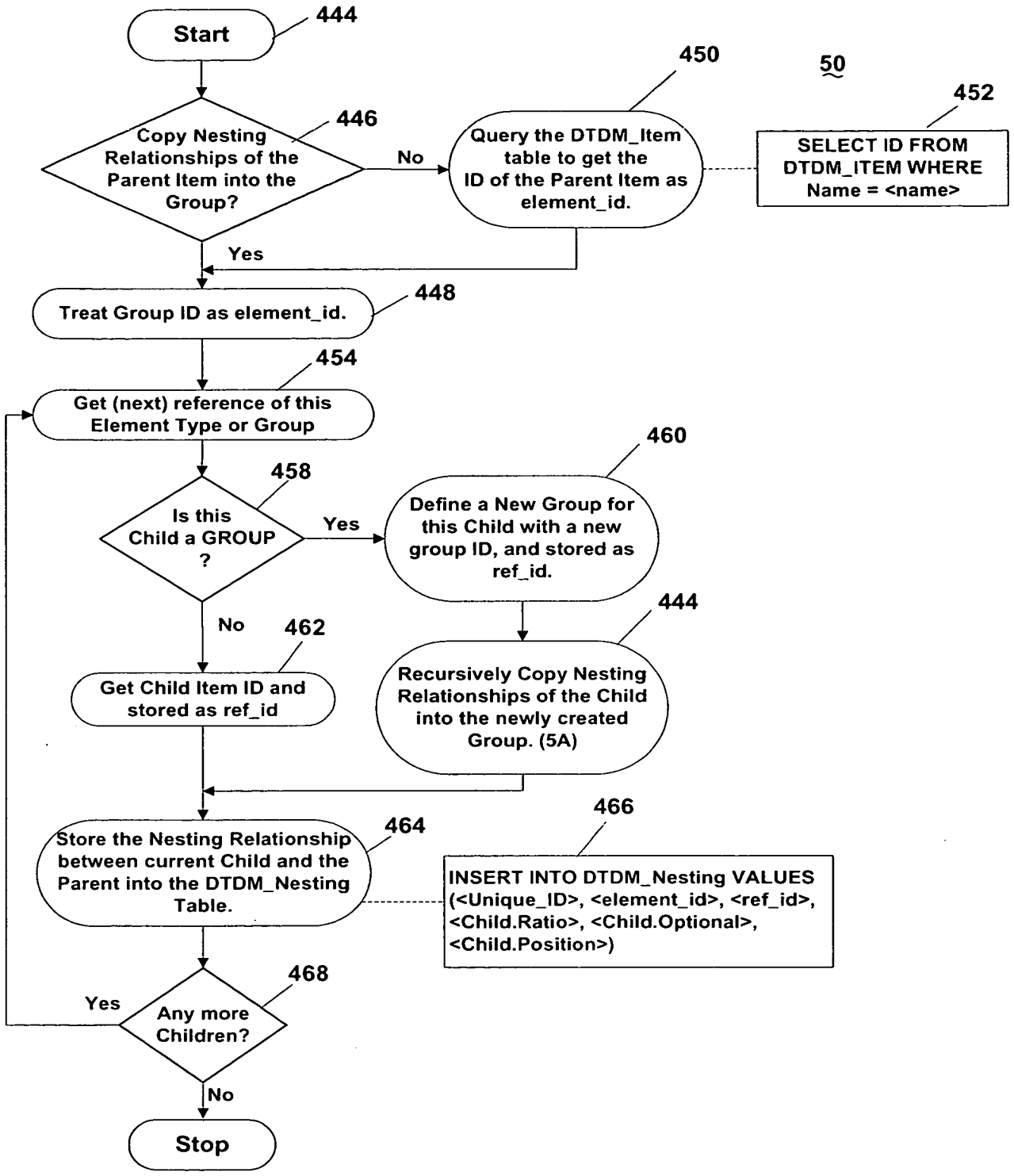
Start
444
Copy Nesting Relationships of the Parent Item into the Group?
446
No
Query the DTDM_Item table to get the ID of the Parent Item as element_id.
450
50
SELECT ID FROM DTDM_ITEM WHERE Name = <name>
452
Yes
Treat Group ID as element_id.
448
Get (next) reference of this Element Type or Group
454
Is this Child a GROUP ?
458
Yes
Define a New Group for this Child with a new group ID, and stored as ref_id.
460
No
Get Child Item ID and stored as ref_id
462
Recursively Copy Nesting Relationships of the Child into the newly created Group. (5A)
444
Store the Nesting Relationship between current Child and the Parent into the DTDM_Nesting Table.
464
INSERT INTO DTDM_Nesting VALUES (<Unique_ID>, <element_id>, <ref_id>, <Child.Ratio>, <Child.Optional>, <Child.Position>)
466
Yes
Any more Children?
468
No
Stop

Fig. 5A

SCANNED IN C.S. FIG.
CLASS SUBCLASS

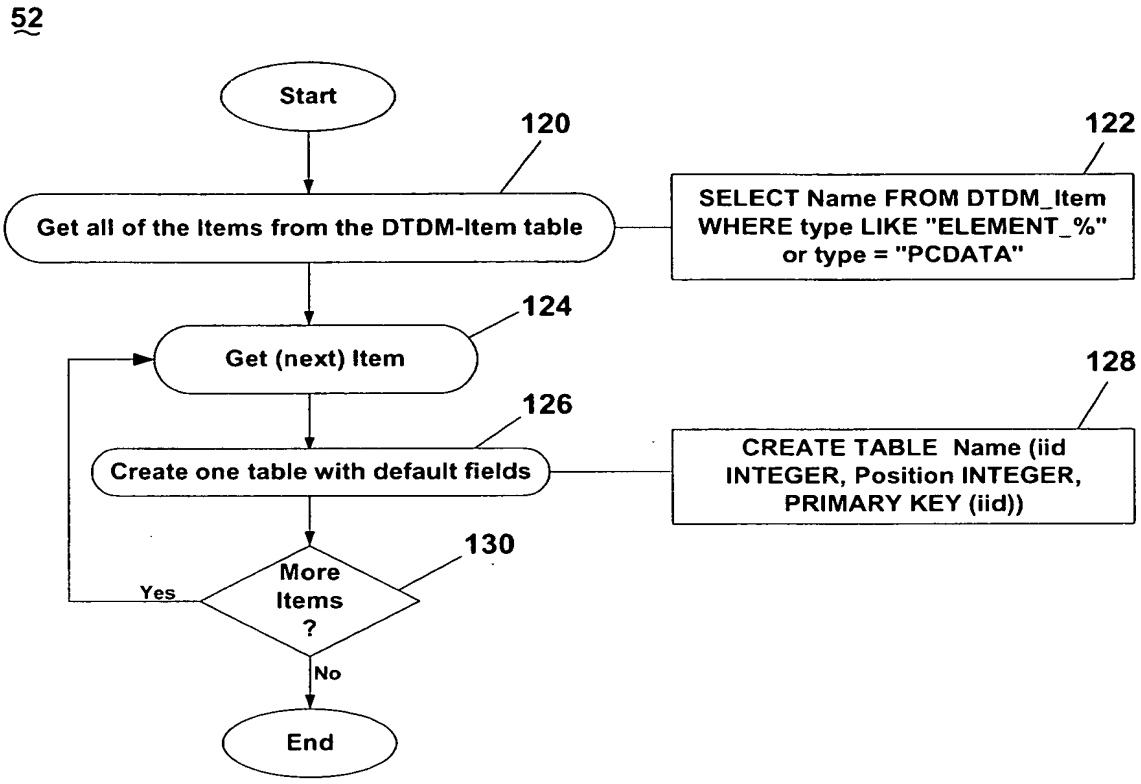

Fig. 6

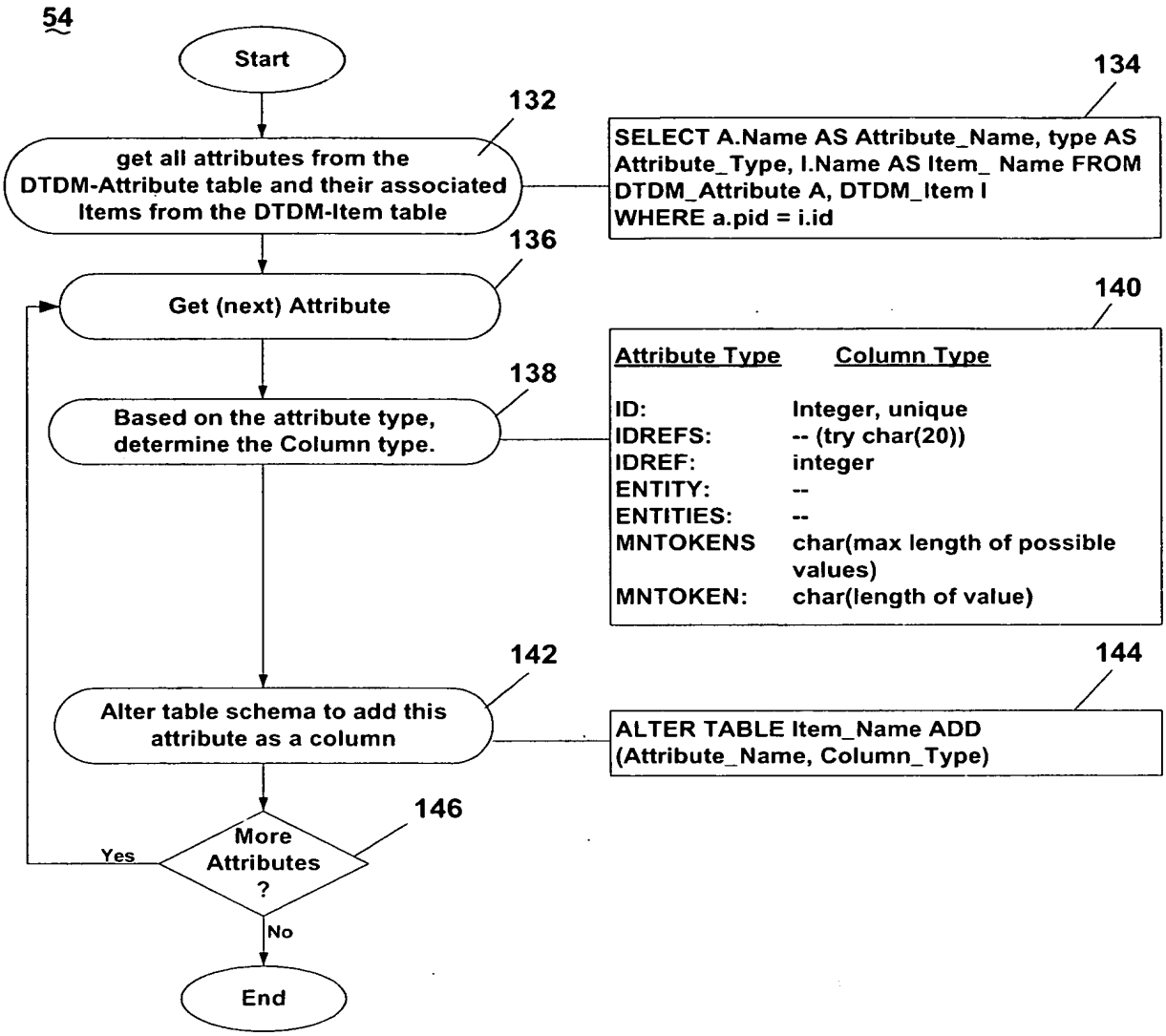
54
Start
132
get all attributes from the DTDM-Attribute table and their associated Items from the DTDM-Item table
134
SELECT A.Name AS Attribute_Name, type AS Attribute_Type, I.Name AS Item_ Name FROM DTDM_Attribute A, DTDM_Item I WHERE a.pid = i.id
136
Get (next) Attribute
140
Attribute Type Column Type
ID: Integer, unique
IDREFS: -- (try char(20))
IDREF: integer
ENTITY: --
ENTITIES: --
MNTOKENS char(max length of possible values)
MNTOKEN: char(length of value)
138
Based on the attribute type, determine the Column type.
142
Alter table schema to add this attribute as a column
144
ALTER TABLE Item_Name ADD (Attribute_Name, Column_Type)
146
More Attributes ?
Yes
No
End

Fig. 7

TO BE PUBLISHED WITH FIG.
SUBCLASS

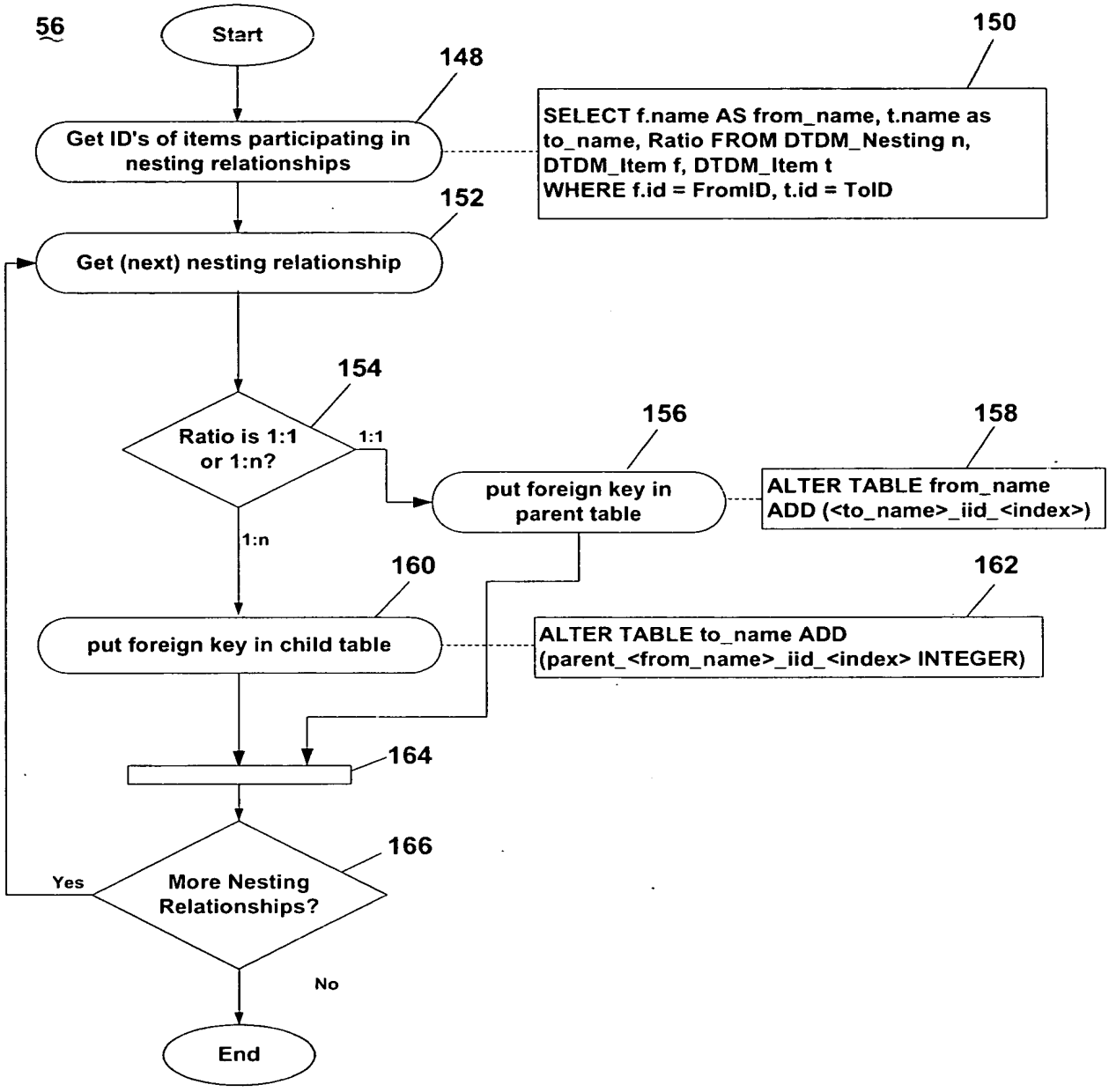

Fig. 8

APPROVED FOR FIG.
CLASS SUBCLASS

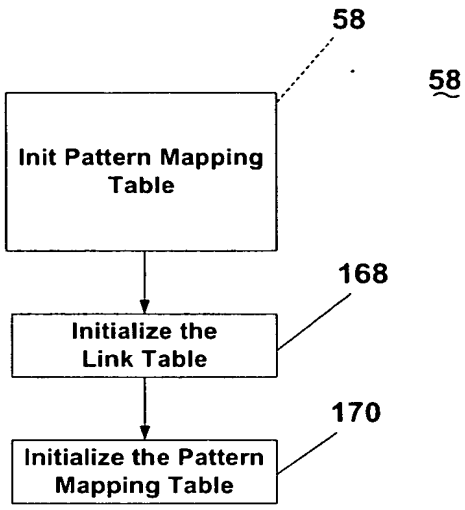

Fig. 9

Docket No. 10020459-1

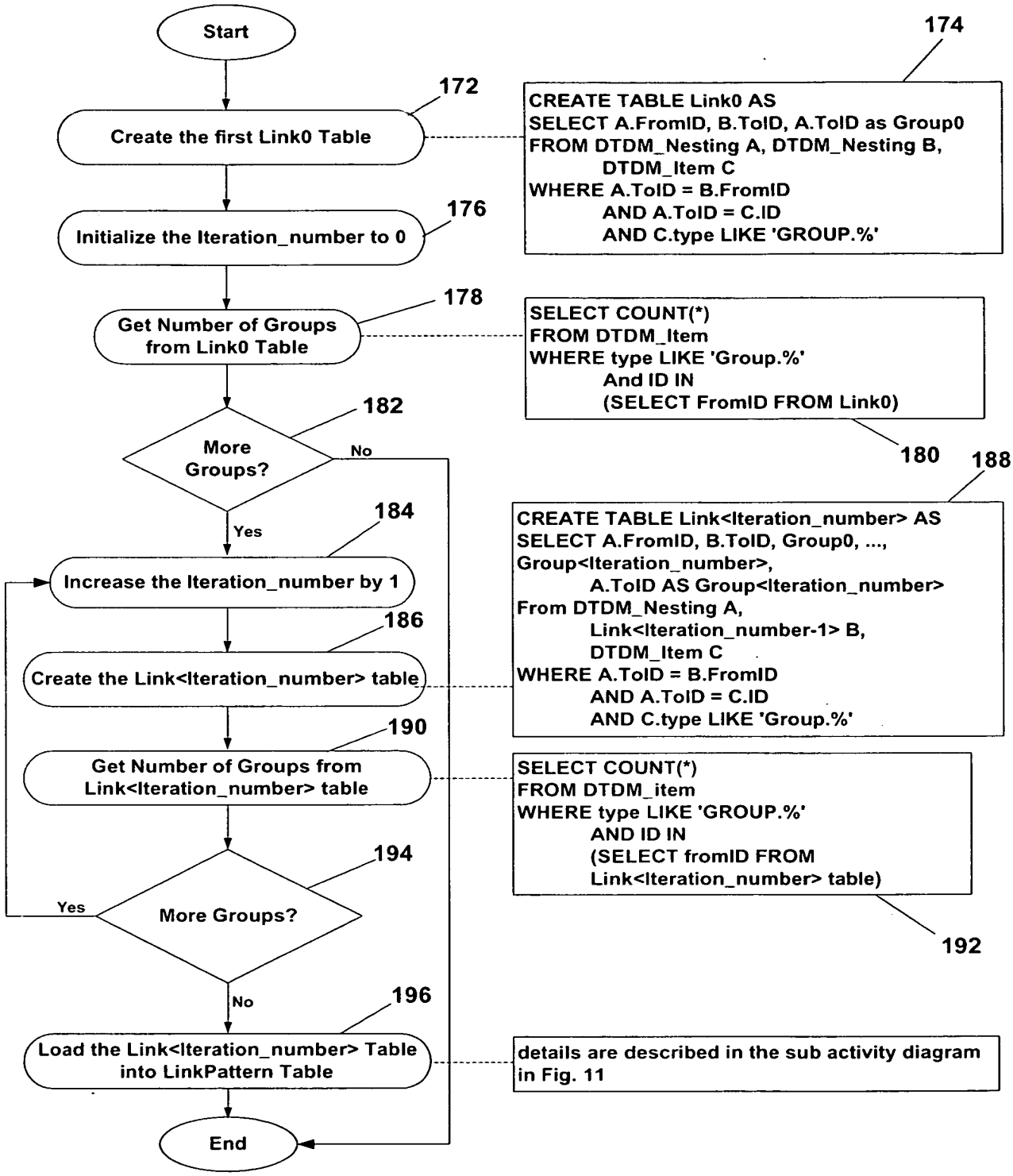
Start
Create the first Link0 Table
172
CREATE TABLE Link0 AS
SELECT A.FromID, B.ToID, A.ToID as Group0
FROM DTDM_Nesting A, DTDM_Nesting B,
DTDM_Item C
WHERE A.ToID = B.FromID
AND A.ToID = C.ID
AND C.type LIKE 'GROUP.%'
174
Initialize the Iteration_number to 0
176
Get Number of Groups from Link0 Table
178
SELECT COUNT(*)
FROM DTDM_Item
WHERE type LIKE 'Group.%'
And ID IN
(SELECT FromID FROM Link0)
180
More Groups?
182
No
Yes
Increase the Iteration_number by 1
184
Create the Link<Iteration_number> table
186
CREATE TABLE Link<Iteration_number> AS
SELECT A.FromID, B.ToID, Group0, ...,
Group<Iteration_number>,
A.ToID AS Group<Iteration_number>
From DTDM_Nesting A,
Link<Iteration_number-1> B,
DTDM_Item C
WHERE A.ToID = B.FromID
AND A.ToID = C.ID
AND C.type LIKE 'Group.%'
188
Get Number of Groups from Link<Iteration_number> table
190
SELECT COUNT(*)
FROM DTDM_item
WHERE type LIKE 'GROUP.%'
AND ID IN
(SELECT fromID FROM
Link<Iteration_number> table)
192
More Groups?
194
Yes
No
Load the Link<Iteration_number> Table into LinkPattern Table
196
details are described in the sub activity diagram in Fig. 11
End

Fig. 10

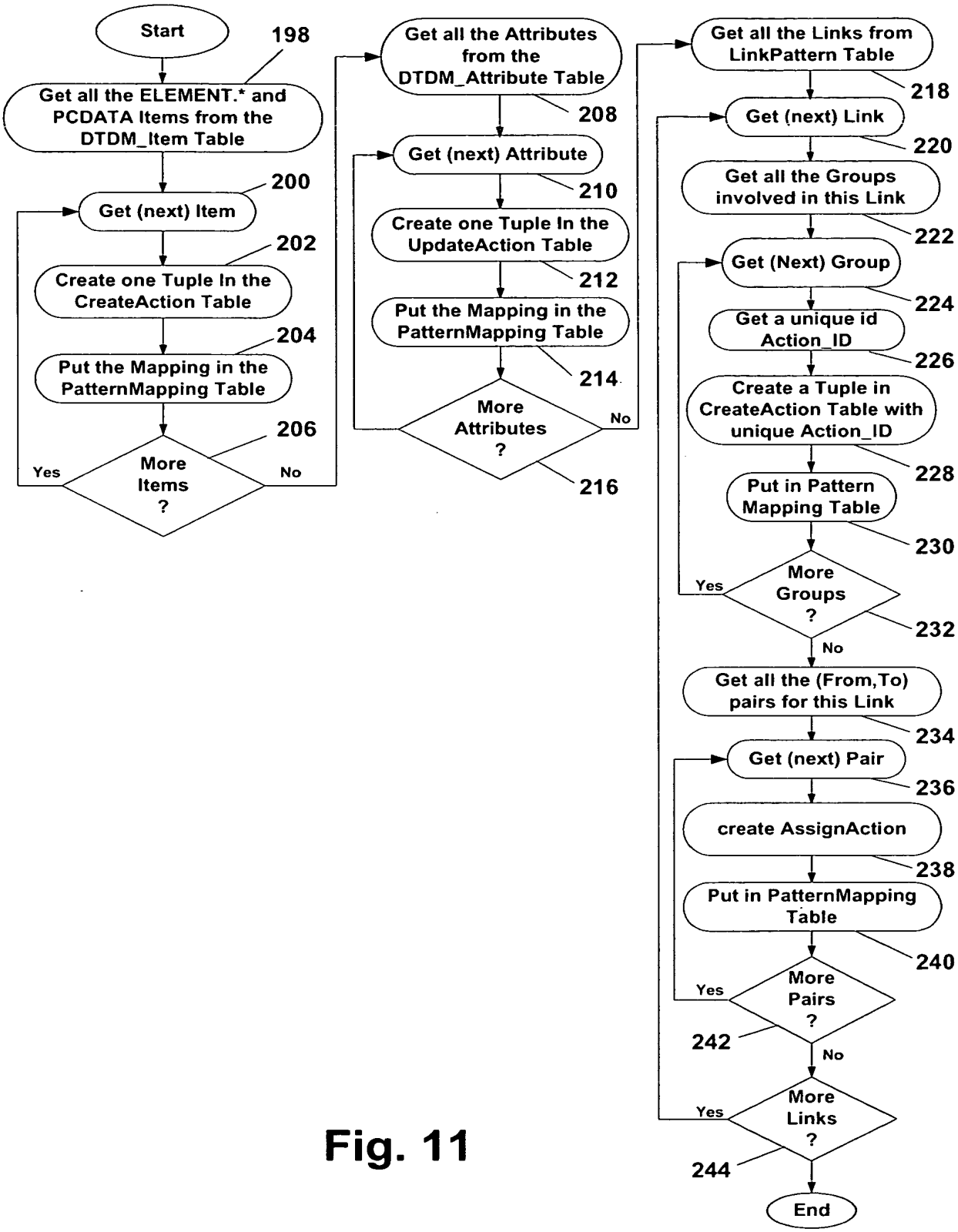
Start
198
Get all the ELEMENT.* and PCDATA Items from the DTDM_Item Table
200
Get (next) Item
202
Create one Tuple In the CreateAction Table
204
Put the Mapping in the PatternMapping Table
206
More Items ?
Yes
No
Get all the Attributes from the DTDM_Attribute Table
208
Get (next) Attribute
210
Create one Tuple In the UpdateAction Table
212
Put the Mapping in the PatternMapping Table
214
More Attributes ?
No
216
Get all the Links from LinkPattern Table
218
Get (next) Link
220
Get all the Groups involved in this Link
222
Get (Next) Group
224
Get a unique id Action_ID
226
Create a Tuple in CreateAction Table with unique Action_ID
228
Put in Pattern Mapping Table
230
More Groups ?
Yes
232
No
Get all the (From,To) pairs for this Link
234
Get (next) Pair
236
create AssignAction
238
Put in PatternMapping Table
240
More Pairs ?
Yes
242
No
More Links ?
Yes
244
End

Fig. 11

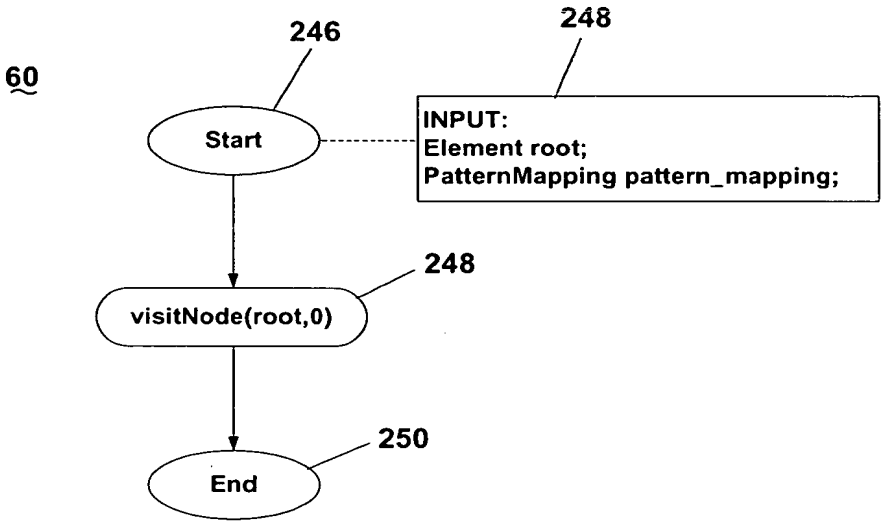
60
246
248
Start
INPUT:
Element root;
PatternMapping pattern_mapping;
248
visitNode(root,0)
250
End

Fig. 12

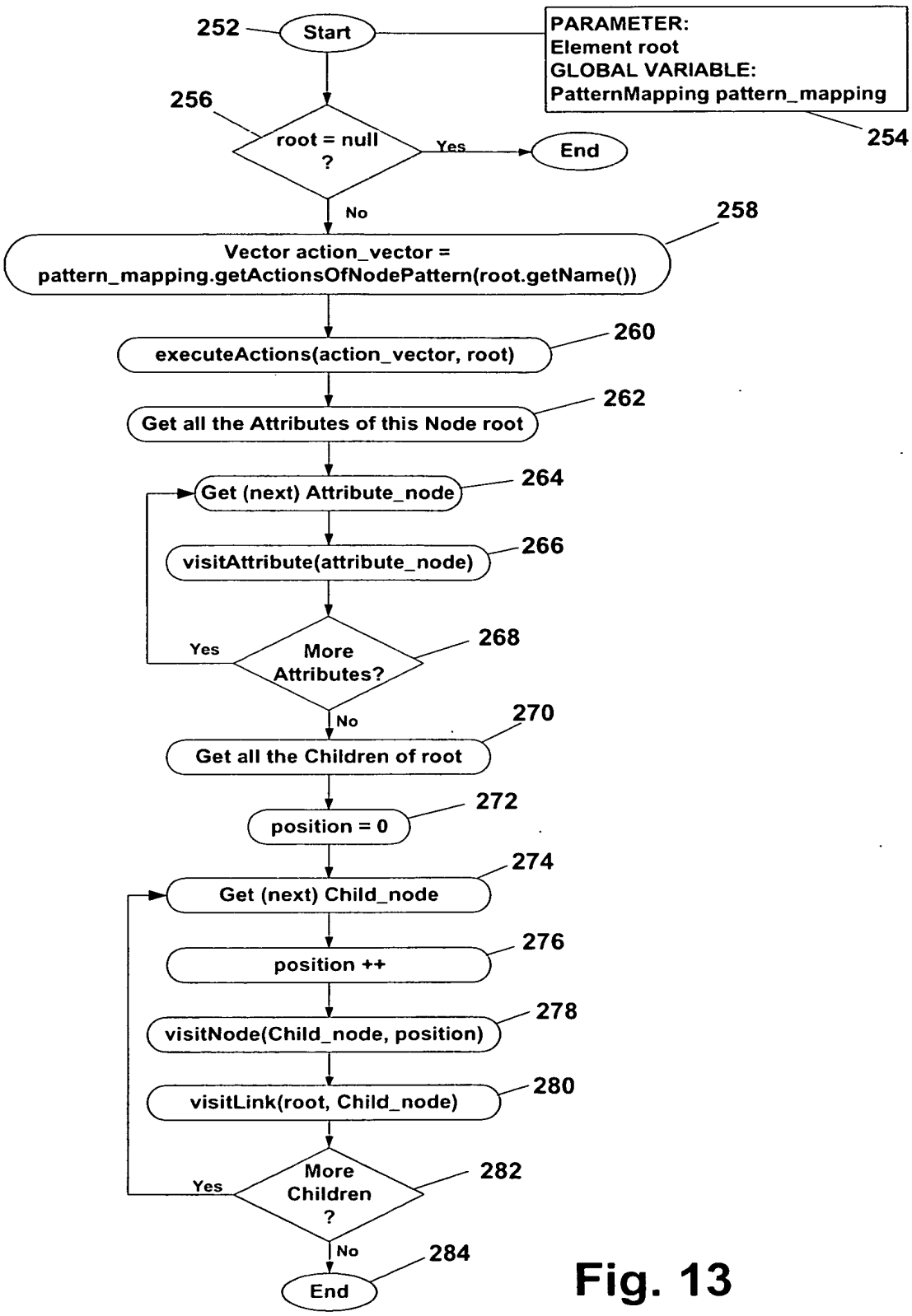
252
Start
PARAMETER:
Element root
GLOBAL VARIABLE:
PatternMapping pattern_mapping
254
256
root = null ?
Yes
End
No
258
Vector action_vector = pattern_mapping.getActionsOfNodePattern(root.getName())
260
executeActions(action_vector, root)
262
Get all the Attributes of this Node root
264
Get (next) Attribute_node
266
visitAttribute(attribute_node)
268
Yes
More Attributes?
No
270
Get all the Children of root
272
position = 0
274
Get (next) Child_node
276
position ++
278
visitNode(Child_node, position)
280
visitLink(root, Child_node)
282
Yes
More Children ?
No
284
End

Fig. 13

T04T20" 259E8760

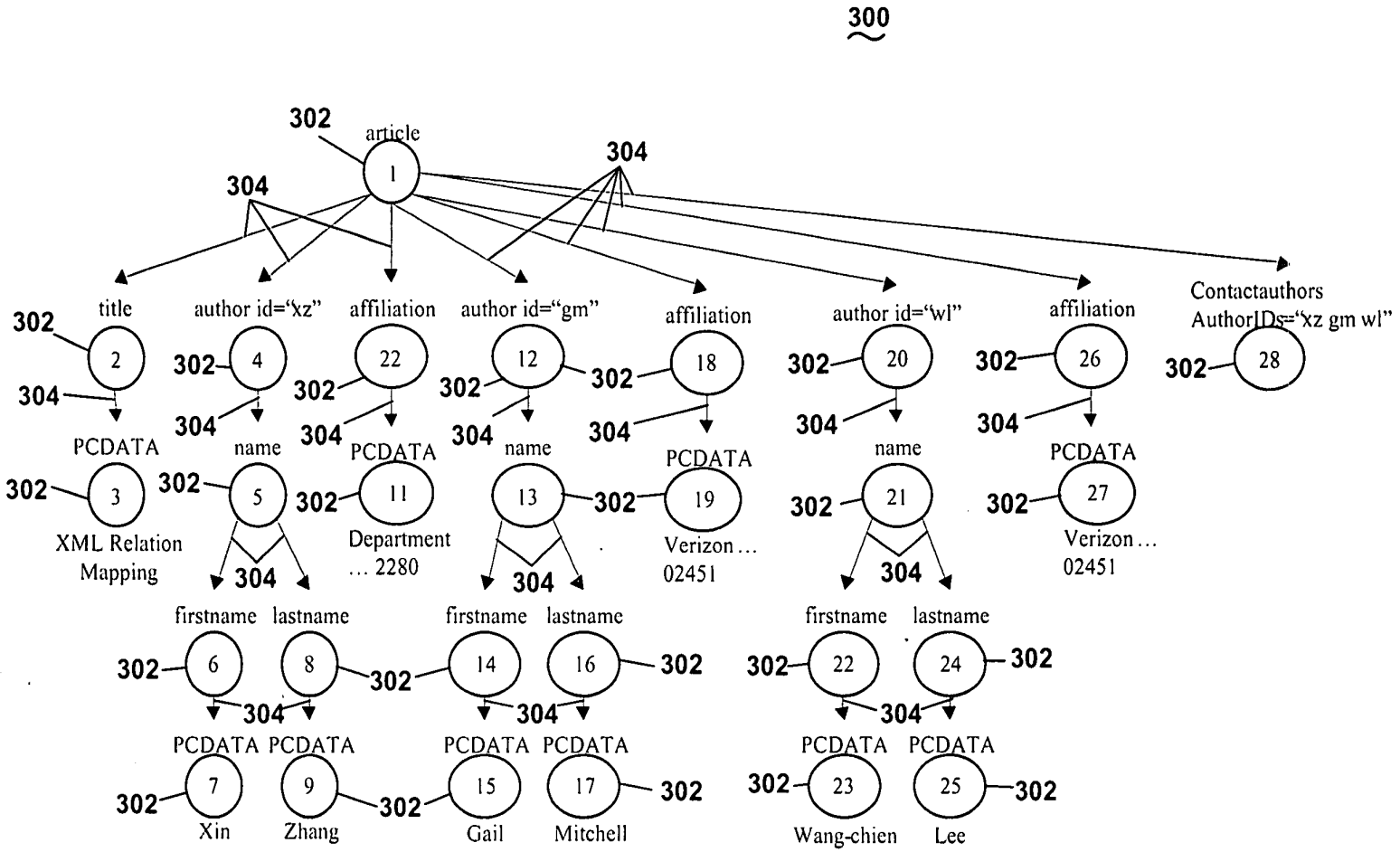

Fig. 14